A restart algorithm for computing fixed points without an extra dimension

Talman, A.J.J.; van der Laan, G.

Published in:
Mathematical Programming

Publication date:
1979

Mathematical Programming 17 (1979) 74–84.
North-Holland Publishing Company

A RESTART ALGORITHM FOR COMPUTING FIXED POINTS WITHOUT AN EXTRA DIMENSION

G. van der LAAN and A.J.J. TALMAN

Free University, Amsterdam, The Netherlands

Received 2 June 1978
Revised manuscript received 27 December 1978

An algorithm to compute a fixed point of an upper semicontinuous point to set mapping using a simplicial subdivision is introduced. The new element of the algorithm is that for a given grid it does not start with a subsimplex but with one (arbitrary) point only; the algorithm will terminate always with a subsimplex. This subsimplex yields an approximation of a fixed point and provides the starting point for a finer grid. Some numerical results suggest that this algorithm converges more rapidly than the known algorithms. Moreover, it is very simple to implement the algorithm on the computer.

Key words: Fixed Points, Triangulation, Grid Size, Labelling Rule.

1. Introduction

Many algorithms based on simplicial subdivisions have been introduced for computing a fixed point of an upper semicontinuous point to set mapping on the unit simplex, cf. Scarf [11, 12], Eaves [2, 3], Merrill [8], Kuhn and MacKinnon [5], and Shapley [13].

Scarf's algorithm is characterised by a particular regular subdivision which is kept fixed throughout the algorithm and by a start in a corner of the unit simplex. In Eaves' algorithm the grid size of the subdivision is automatically increased in the course of the algorithm with a factor of incrementation of two. The Sandwich method of Merrill and Kuhn and MacKinnon operates with a particular regular subdivision but it can start anywhere, so we may apply the algorithm succesively for a sequence of grids with increasing size using the approximate solution for a certain grid in the sequence in the next grid. A survey of these algorithms is given by Lüthi [7] and Todd [14].

In this paper we introduce an algorithm which has the same advantages as the Sandwich method of Merrill and Kuhn and MacKinnon, but it does not use a set of artificial labelled points. Furthermore, it differs from all the algorithms mentioned above by the fact that it does not start in a certain grid with a full dimensional subsimplex but with one point only. It terminates with a full dimensional subsimplex providing the starting point for a new application of the algorithm for a finer grid. Successive grid refinements will produce a sequence of

approximate fixed points which converge to a fixed point of the mapping. Analogous to the Sandwich method the factor of incrementation may be of any size and even different at each state.

In Section 2 some preliminaries are given. Section 3 gives the description of the method for integer labelling for continuous functions on the unit simplex. In Section 4 it is proved that a completely labelled full dimensional subsimplex is always found. We will discuss the application of the algorithm on the unit simplex for vector labelling in Section 5. Numerical results are given in Section 6 and conclusions are drawn in Section 7.

2. Preliminaries

Let $S^n = \{x \in R^n \mid x_i \geq 0 \text{ for all } i \text{ and } \sum_{i=1}^n x_i = 1\}$ be the $(n-1)$-dimensional unit simplex.

Define $S^n(m)$ as the set of points of S^n induced by the regular grid of size m, i.e. the elements of $S^n(m)$ are the points x such that $x = (y_1/m, \ldots, y_n/m)$, where y_i is a non-negative integer and $\sum_{i=1}^n y_i = m$. For ease of notation we delete in the following the denominator m.

Define the $n \times n$ matrix Q by

$$Q = \begin{bmatrix} -1 & 0 & \cdots & 0 & 1 \\ 1 & -1 & & 0 & 0 \\ 0 & 1 & \ddots & & \\ \vdots & & \ddots & & \vdots \\ & & & -1 & 0 \\ 0 & 0 & \cdots & 1 & -1 \end{bmatrix}$$

The ith column of Q will be denoted by $q(i)$. Let T be a subset of $I_n = \{1, 2, \ldots, n\}$ with $|T| = t$, γ^T some permutation of T, and $\{v^j, j = 0, \ldots, t\}$ a set of points of $S^n(m)$, such that $v^j = v^{j-1} + q(\gamma_j)$ for $j = 1, \ldots, t$. The convex hull of the points $v^0, \ldots, v^t$ is called a t-dimensional face and is denoted by $\tau(v^0, \ldots, v^t)$. A sub simplex is an $(n-1)$-dimensional face. Observe that every t-dimensional face $(t \leq n-2)$ has a unique representation (v^0, γ^T), whereas a subsimplex has n representations.

Let f be a continuous function from S^n to S^n. From Brouwer's theorem [1] we know there exists a fixed point x^*, i.e. $f(x^*) = x^*$. To compute x^*, the points of S^n receive a label induced by the function f. In the case of integer labelling the labels are determined by the following rule;

$l(x) = i$ if i is the lowest index with $x_i > 0$ and

$$f_i(x) - x_i \leq f_k(x) - x_k \text{ for all } k \in I_n.$$

Note that this labelling rule is proper.

A face will be called completely labelled if all its vertices have a different label. A completely labelled subsimplex is a good approximation of a fixed point of the function f.

Using vector labelling a point x of S^n receives a label l defined by

$$l(x) = f(x) - x + e$$

where $e = (1, \dots, 1)$. A t-dimensional face $\tau(w^0, w^1, \dots, w^t)$ $(t \leqslant n-1)$ will be called completely labelled if the linear system

$$\sum_{i=0}^{t} \lambda_i l(w^i) + \sum_{j=1}^{n-t-1} \mu_j e(\pi_j) = e$$

where $e(i)$ is the ith column of the $n \times n$ identity matrix and where $(\pi_1, \dots, \pi_{n-t-1})$ is a permutation of the elements not in T, has a nonnegative solution λ_i^*, $i = 0, \dots, t$, and μ_j^*, $j = 1, \dots, n-t-1$.

3. Description of the algorithm

We give only the description for integer labelling. Some remarks about vector labelling are made in Section 5.

For a given gridsize m the algorithm starts in an arbitrary point v^0 of $S^n(m)$ by computing its label $l(v^0)$. Note that $\tau(v^0)$ is a completely labelled zero dimensional face. From this face on the algorithm will generate a path of faces, which terminates with a completely labelled subsimplex. To describe the procedure we assume the algorithm generates a subset T of I_n with $|T| = t$, $t \leqslant n-1$, a permutation γ^T, a point w^0 of $S^n(m)$ and a non-negative n-dimensional vector R, such that $R_j = 0$, $j \notin T$, $w^0 = v^0 + \sum_{j=1}^{n} R_j q(j)$ and $\tau(w^0, w^1, \dots, w^{t-1})$ where $w^j = w^{j-1} + q(\gamma_j)$ for $j = 1, \dots, t-1$, is a completely labelled $(t-1)$-dimensional face with $l(w^j) \in T$, for $j = 0, \dots, t-1$. Note that this assumption is satisfied for

$$w^0 = v^0, \qquad T = \{l(v^0)\}, \qquad \gamma^T = l(v^0), \qquad R = (0, \dots, 0).$$

The completely labelled $(t-1)$-dimensional face is now extended to a t-dimensional face by adding the vertex $w^t = w^{t-1} + q(\gamma_t)$. The label $l(w^t)$ is computed. Then either $l(w^t)$ is an element of T or it is not. In the latter case we have a t-dimensional completely labelled face for which the above mentioned assumption is satisfied for $T_1 = T \cup \{l(w^t)\}$, $\gamma^{T_1} = (\gamma^T, l(w^t))$, whereas w^0 and R do not change.

If $l(w^t) \in T$, then $l(w^t) = l(w^j)$ for some $j \in \{0, 1, \dots, t-1\}$. Then we adapt the vector w^0, the permutation γ^T and the vector R according to Table 1, i.e. w^j is replaced by a new vertex, say $\bar{w}$.

Table 1
s is the index of the vector which must be replaced

	w^0 becomes	γ^T becomes	R becomes
$s = 0$	$w^0 + q(\gamma_1)$	$(\gamma_2, \dots, \gamma_t, \gamma_1)$	$R + e(\gamma_1)$
$1 \leq s \leq t-1$	w^0	$(\gamma_1, \dots, \gamma_{s-1}, \gamma_{s+1}, \gamma_s, \gamma_{s+2}, \dots, \gamma_t)$	R
$s = t$	$w^0 - q(\gamma_t)$	$(\gamma_t, \gamma_1, \dots, \gamma_{t-1})$	$R - e(\gamma_t)$

If $l(\bar{w}) \in T$, $l(\bar{w})$ must be equal to $l(w^i)$ for some $i \neq j$. As long as R remains nonegative and labels in T are found, the algorithm continues by making a replacement step and computing the label of the new vertex. Doing so, the algorithm generates a path of adjacent t-dimensional faces having t different labels. This implies that we are operating in an n-dimensional space with $n-t$ independent linear restrictions.[1] In other words the algorithm operates actually with subsimplices in a t-dimensional subspace. In the next section we will prove that this implies that after a finite number of replacement steps either the algorithm finds a label k not in T or R_j becomes negative for certain j in T. In the first case the above mentioned assumption is satisfied for $T_1 = T \cup \{k\}$, $\gamma^{T_1} = (\tilde{\gamma}^T, k)$, $w^0 = \tilde{w}^0$ and $R = \tilde{R}$, if $\tilde{\gamma}^T$, $\tilde{w}^0$ and $\tilde{R}$ are generated in the last replacement step. Then the algorithm continues with the new vertex $w^{t+1} = \tilde{w}^t + q(k)$ by computing its label. The second case can only happen after adapting certain $\tilde{w}^0$, $\tilde{R}$ and $\tilde{\gamma}^T$ when $s = t$. Hence, R_j becomes negative for $j = \tilde{\gamma}_t$. Instead of removing now the last vertex $\tilde{w}^t$, this point is deleted, $R_{\tilde{\gamma}_t}$ is set equal to zero, T becomes $T \setminus \{\tilde{\gamma}_t\}$, γ^T becomes $(\tilde{\gamma}_1, \dots, \tilde{\gamma}_{t-1})$ and the algorithm goes on by replacing the vector with label $\tilde{\gamma}_t$, the label which is deleted.

Consequently, starting with a $(t-1)$-dimensional completely labelled face, the algorithm either finds a t-dimensional completely labelled face or continues after a finite number of replacement steps with a $(t-1)$-dimensional face with $t-1$ different labels. To approximate a fixed point we start the algorithm in an *arbitrary point* v^0 with $w^0 = v^0$, $T = \{l(v^0)\}$, $\gamma^T = l(v^0)$ and $R = (0, \dots, 0)$. In the next section it is proved that the algorithm always terminates with a completely labelled subsimplex. To illustrate the algorithm a path of adjacent faces is drawn for $n = 3$ and $m = 9$ in Fig. 1.

The algorithm starts in $v^0 = (3, 5, 1)$. The points $(2, 6, 1)$ and $(1, 7, 1)$ are the vertices of the first completely labelled one dimensional face. Then the algorithm continues with $\tau((2, 6, 1), (1, 7, 1), (1, 6, 2))$. In $\tau((3, 4, 2), (3, 3, 3), (2, 4, 3))$ point $(2, 4, 3)$ has to be deleted and $(3, 4, 2)$ must be replaced, i.e. the algorithm goes on with one dimensional faces. The points (3, 2,4) and (3, 1, 5) are again the vertices

[1] To prove this, assume without loss of generality that n is not in T. Then there is a partition $T_1, \dots, T_l$ of T, such that $s \in T_i$ implies either $s+1 \in T_i$ or $s+1 \notin T$. Let $s_i = \max\{s \mid s \in T_i\}$, then for each new vector x used in the algorithm $\sum_{j \in T_i} x_j + x_{s_i+1} = c_i$ for $i = 1, \dots, l$, where c_i is constant. Furthermore x_j is constant for $j \notin \bigcup_{i=1}^{l} (T_i \cup \{s_i + 1\})$, which are $n - (t+l)$ constants. Thus together there are $l + n - (t+l) = n - t$ independent restrictions on the components of x.

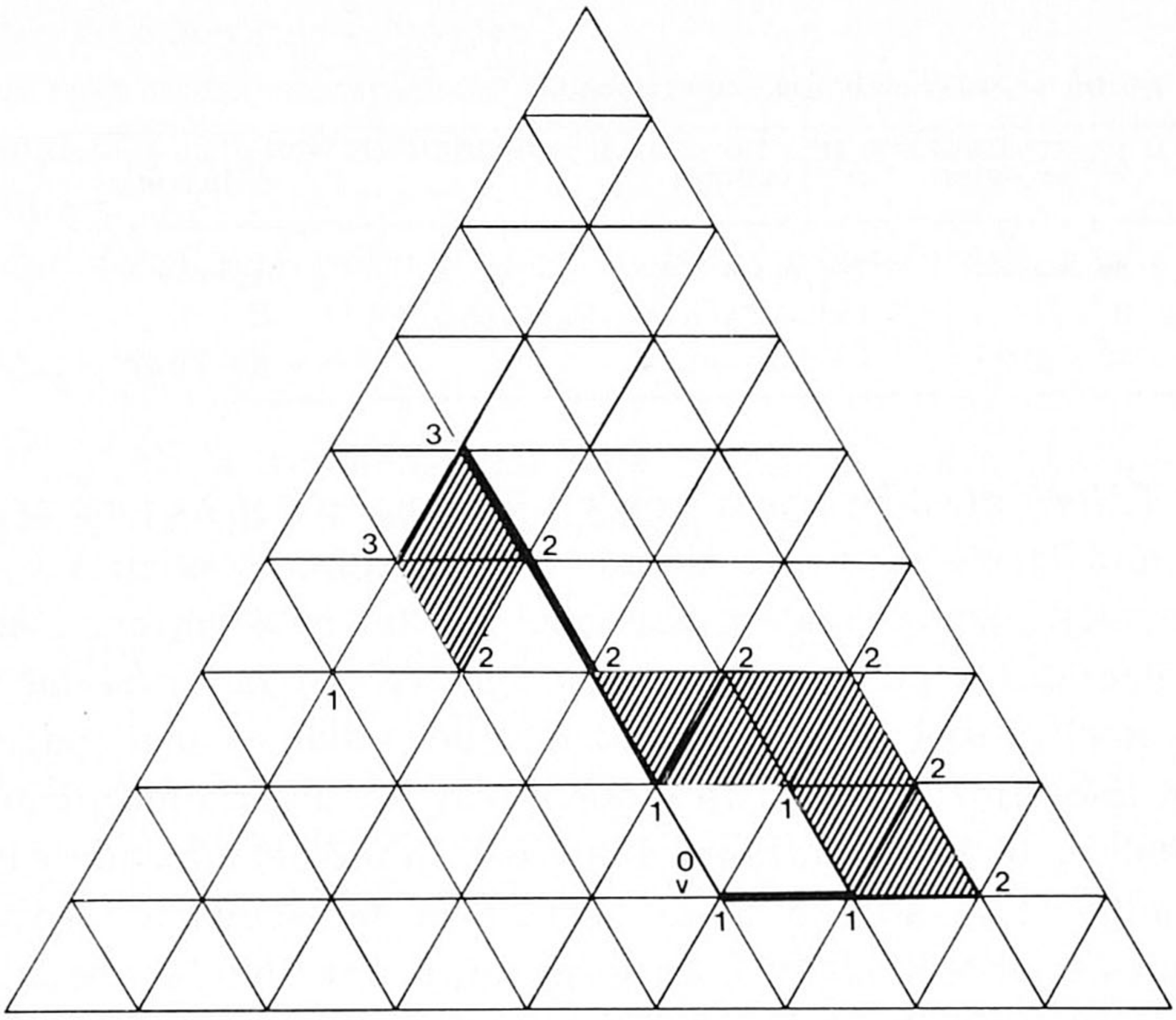

Fig. 1. $n = 3$, $m = 9$, $v^0 = (3, 5, 1)$.

of a completely labelled one dimensional face. Finally, $(5, 1, 3)$, $(4, 2, 3)$ and $(4, 1, 4)$ are the vertices of a completely labelled subsimplex.

The above described method makes it possible to approximate a fixed point using successively grids $S^n(k_1), S^n(k_2), \ldots$ with an increasing sequence of grid sizes, without a triangulation of $S^n \times [1, \infty]$ (Eaves [3]) or artificial labelling [8, 5]. The completely labelled subsimplex of $S^n(k_i)$ found by the algorithm may be used as prior information in the next grid $S^n(k_{i+1})$.

4. Geometrical interpretations and theoretical termination

Before we prove that the algorithm finds always a completely labelled subsimplex, we give some geometrical clarifications. Starting in a point v^0 of $S^n(m)$ let us define a subregion $A(T)$ for any subset T of I_n by

$$A(T) = \{x \in S^n(m) \mid x = v^0 + \sum_{j \in T} k_j q(j), \text{ for non-negative integers } k_j,\ j \in T\}.$$

Note that $A(\emptyset) = \{v^0\}$. A gridpoint of $S^n(m)$ will be called a proper point of $A(T)$ if it is an element of the subregion $A(T)$ such that $k_j > 0$, for all $j \in T$.

The construction of the algorithm is such that if it generates certain T, γ^T, w^0 and R, then $R_i = 0$, $i \notin T$, and the point w^0 is equal to $v^0 + \sum_{j=1}^{n} R_j q(j)$. Hence, $w^0, \ldots, w^t$ are all elements of $A(T)$. Note that w^t is always a proper element of $A(T)$. Also, when w^s is proper for some s, $0 \leq s \leq t-1$, w^i is proper for all $i \geq s$.

These features are illustrated for $n = 3$ in Fig. 2. If in the course of the algorithm the subsimplex E is generated, then $T = \{1, 2\}$, $\gamma^T = (2, 1)$, whereas R_1 and R_2 are positive, i.e. the vertices of E are all proper elements of $A(1, 2)$. Moreover, either two vertices of E have the same label, viz. an element of T, while the third vertex has a label equal to the other element of T, or all the three vertices have a different label. In the latter case the vertex with label 3 must be the last generated vertex and the algorithm terminates.

If the subsimplex F is generated, then $T = \{1, 3\}$, $\gamma^T = (3, 1)$, and R_3 is positive, whereas R_1 is equal to zero (of course $R_2 = 0$). When vertex a, the only proper point of the face, should be removed, R_1 becomes negative. This implies that the vertex a has to be deleted and that the vertex b (which must have label 1) must be removed. The algorithm goes on with a one-dimensional face with vertices c and d, i.e. the algorithm continues with $T = \{3\}$, $\gamma^T = (3)$, $w^0 = c$ and R_3 positive, implying that both c and d are proper elements of $A(3)$. In general a component of R becomes negative if and only if w^t is the only proper point of $A(T)$ and has to be removed, for the replacement step would imply in this case a change from a t-dimensional face with vertices in $A(T)$ to a t-dimensional face with vertices in an "adjacent" region $A(T \cup \{k\} \setminus \{j\})$ for some $j \in T$ and $k \notin T$. This is avoided by deleting the only proper point w^t. The algorithm is continued with a $(t-1)$-dimensional face with vertices in $A(T) \setminus \{j\})$ by removing the vertex with label j, whereas T is set equal to $T \setminus \{j\}$.

Now we prove the algorithm always finds a completely labelled subsimplex. First it is shown that the replacement step between two t-dimensional faces is always feasible. Clearly, if a vertex has to be removed, the replacement step will produce a new feasible vertex except in the case that the remaining vertices are all points on the same side of the unit simplex. So, let for some s, w^s be the only

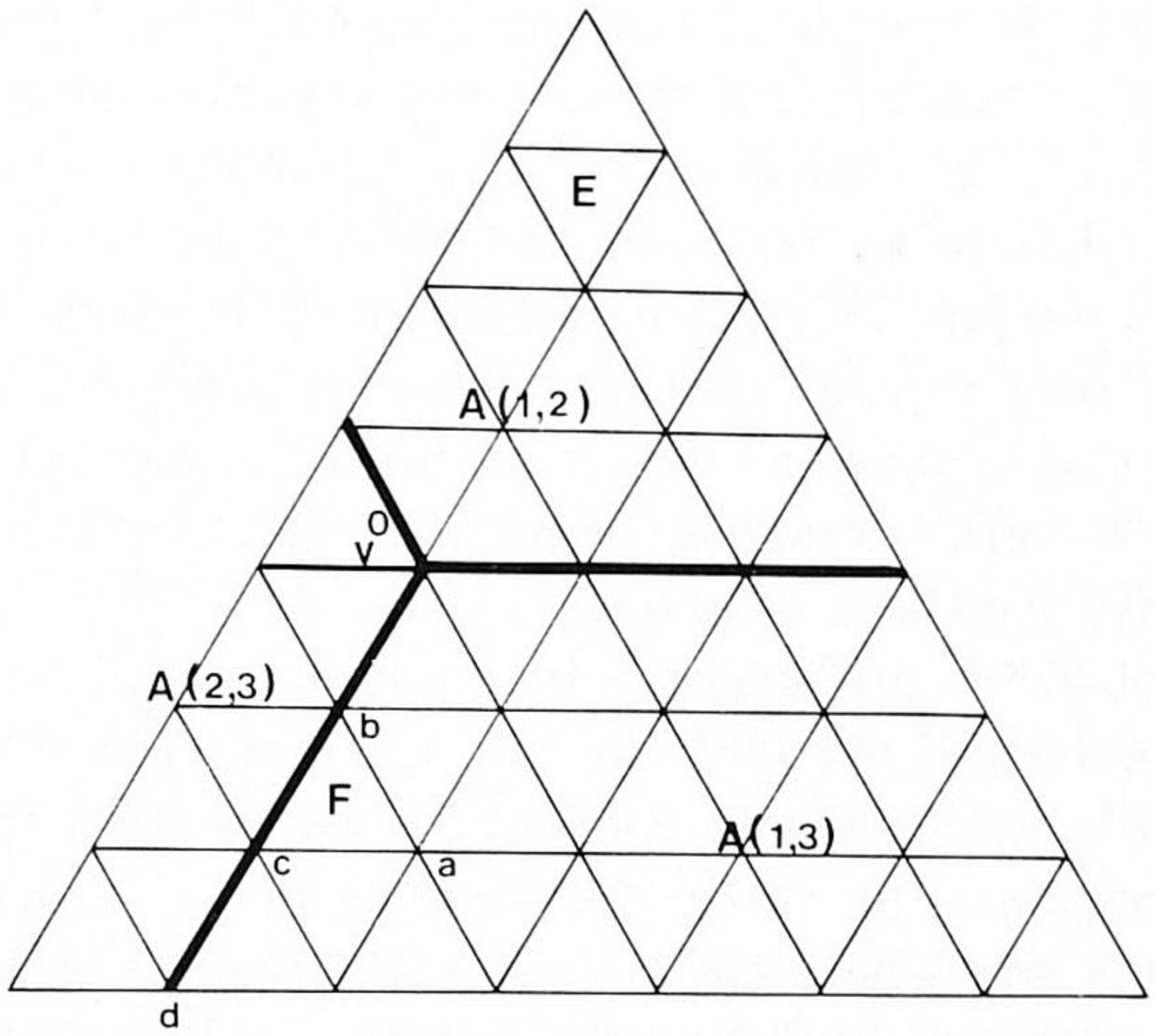

Fig. 2. $n = 3$, $m = 7$, $v^0 = (3, 1, 3)$.

vertex not on the jth side of S^n, for some j, and let $w^0, \ldots, w^{s-1}, w^{s+1}, \ldots, w^t$ be the remaining vertices all having the jth component equal to zero. If the starting point v^0 is not on the jth side, j must be an element of T, i.e. at least one of the vertices of the face must have label j. Since we have a proper labelling, w^s is the only vertex with label j, i.e. w^s cannot be removed. If, however, v^0 is on the jth side of the unit simplex, either one of the vertices has label j, which is identical to the case mentioned just above, or j is not an element of T. Then $s = t$ and the γ_tth component of R becomes negative, since w^t is the only proper vertex in $A(T)$. Consequently, the vertex w^t is deleted and the algorithm continues as described above with a $(t-1)$-dimensional face having vertices only in $A(T) \setminus \{\gamma_t\}$). All this together proves that all replacement steps are feasible. Moreover, it is clear that the replacement step between two equal dimensional faces is unique (see [12, p.43]).

Next we prove that the start of the algorithm is unique. As shown above, the algorithm starts actually with a one-dimensional face with vertices v^0 and $v^1 = v^0 + q(l(v^0),\ T = \{l(v^0)\},\ \gamma^T = l(v^0)$ and $R = (0, \ldots, 0)$. Note that v^1 is a proper point of $A(T)$ but v^0 is not. If $l(v^1) \neq l(v^0)$, there is a unique extension to a two dimensional face which we will prove below. If not by removing v^1 the component of R on place $l(v^0)$ becomes negative, indicating that v^1 has to be deleted and v^0 must be removed. Since v^0 is a zero-dimensional face this is impossible. Therefore v^0 must be removed and the start of the algorithm is unique. It remains to prove that the algorithm provides a unique feasible change from a t-dimensional (completely labelled) face to a $(t+1)$-dimensional face $(1 \leq t \leq n-2)$ or to a $(t-1)$-dimensional $(2 \leq t \leq n-1)$. In the first case, let $w^0, \ldots, w^t$ be a completely labelled t-dimensional face and j the label just found. This implies $R_j = 0$. The face is extended with the vertex $v^{t+1} = v^t + q(j)$. Without loss of generality we assume that this point has a label equal to $l(w^s)$, for some s, $1 \leq s \leq t$. Now we have actually the choice between removing w^s or w^{t+1}. But replacing w^{t+1} means that R_j becomes negative, implying that w^{t+1} has to be deleted. Hence, by replacing w^{t+1} the algorithm returns, which is not admissible. So, w^s has to be removed and the change from a t to a $(t+1)$-dimensional face is unique. Using same arguments the reverse is also true.

All this together demonstrates that the algorithm generates a unique feasible path of faces such that cycling can not occur. Since the number of faces is finite, the algorithm must then terminate with a completely labelled subsimplex (Lemke's argument), as was to be proved.

An interpretation with artificial labelled points can be given, which was independently found by Todd [15]. In this interpretation $S^n \times [0, 1]$ is triangulated. When $\tau(w^0, \ldots, w^t)$ is a face of $A(T)$, we create a subsimplex of the triangulation by adjoining to $\tau \times \{0\}$ the vertices $\hat{e}(\pi_1), \ldots, \hat{e}(\pi_{n-t})$ of $S^n \times \{1\}$, where $\hat{e}(j) = (e(j), 1)$ and where $(\pi_1, \ldots, \pi_{n-t})$ is a permutation of the elements not in T. The point $\hat{e}(j)$ receives the label j, while a point of $S^n \times \{0\}$ receives a proper label induced by f. Now, starting with the unique completely labelled

facet $\hat{e}(1), \ldots, \hat{e}(n)$ of $S^n = \{1\}$ and the vertex $(v^0, 0)$, the algorithm generates a unique path of adjacent subsimplices such that the common facets are completely labelled, and it terminates with a completely labelled facet in $S^n \times \{0\}$. The intersection of the path of subsimplices with $S^n \times \{0\}$ is the sequence of faces obtained by the algorithm described above.

5. Vector labelling

In this section we discuss the application of the method for vector labelling. In case of an upper semicontinuous point to set mapping Φ, $l(x) = f(x) - x + e$, where $f(x)$ is some element of $\Phi(x)$. The algorithm starts again with one gridpoint only, say v^0, together with the system of linear equations $Iy = e$. By pivoting, one of the unit vectors is replaced by $l(v^0)$. When $e(j)$ is eliminated, the algorithm goes on from the one dimensional face $\tau(v^0, v^0 + q(j))$, $T = \{j\}$, $\gamma^T = j$ and $R = (0, \ldots, 0)$, with alternating pivoting and replacement steps. The algorithm has found a completely labelled t-dimensional face $w^0, \ldots, w^t$, as soon as $e(i)$, for some $i \notin T$, is eliminated by a pivot step. Then the point $w^t + q(i)$ is added to the previous face $\tau(w^0, \ldots, w^t)$. If in a replacement step R_j becomes negative for some j, then the last vertex w^t of the face is deleted, a pivot step is made with the unit vector $e(j)$, and the algorithm continues as before, i.e. either a replacement step is made with the vertex w^s, if $l(w^s)$ is eliminated by $e(j)$, or $t(w^0, \ldots, w^{t-1})$ is extended with the vertex $w^{t-1} + e(i)$, if $e(i)$ is eliminated by $e(j)$.

By the same arguments as in the previous section the algorithm terminates with a completely labelled subsimplex.

6. Results

The algorithm was applied to three examples in which equilibrium price vectors in a Walras model are to be computed. The data of these problems are given by Scarf [11] and were also used by Wilmuth [16] and Kuhn and MacKinnon [5].

In Tables 2–4 the cumulative number of iterations for the sequence of regular grids $S^n(n)$, $S^n(n \times f)$, $S^n(n \times f^2), \ldots$ are given, where f is some factor of incrementation. Let $\tau(w^0, \ldots, w^{n-1})$ be the completely labelled subsimplex found by the algorithm in $S^n(n \times f^i)$, $i = 1, 2, \ldots$. Then the starting poin in $S^n(n \times f^{i+1})$ was chosen to be $v^0 = (\sum_{j=0}^{f-1} w^{j(\mathrm{mod}\, n)})/f$. In the first stage the barycenter was always chosen.

For the three examples we applied the algorithm with some modifications. Instead of Q we used $-Q$. For integer labelling the boundary of the simplex was artificial labelled, viz. for all x on the boundary, $l(x) = i$, if i is the lowest index

Table 2
Scarf's problem 1; $n = 5$. The cumulative number of iterations for integer (Z_1) as well as for vector labelling (Z_2). The factor of incrementation is f and the grid size is $n \times f^i$ if $i = j$

f	2		3		4		5	
i	Z_1	Z_2	Z_1	Z_2	Z_1	Z_2	Z_1	Z_2
2	23	25	33	37	30	37	43	35
4	39	38	55	57	55	59	86	53
6	62	51	81	79	82	77	112	79
8	77	68	100	104	116	95	159	102
9	90	79	112	116	132	104	176	111
11	105	100	133	141	151	124		
14	134	134	156	170				
18	166	176						
22	198	216						

Table 3
Scarf's problem 2; $n = 8$

f	2		3		4		5	8
i	Z_1	Z_2	Z_1	Z_2	Z_1	Z_2	Z_2	Z_2
2	50	47	58	56	103	95	91	145
4	72	96	108	100	216	158	128	201
6	115	141	155	126	289	193	174	257
8	183	163	206	180	435	247	223	
9	204	175	222	196	479	278		
11	249	199	290	231				
14	298	251						
18	392	310						

Table 4
Scarf's problem 3; $n = 10$

f	2		3		4		6	8	10
i	Z_1	Z_2	Z_1	Z_2	Z_1	Z_2	Z_2	Z_2	Z_2
3	66	79	112	121	120	97	104	154	173
5	95	121	206	166	219	159	203	212	257
6	119	146	245	185	279	192	241	249	
7	150	164	288	210	368	214	272		
9	199	213	371	277	499	272			
11	247	259	439	337					
14	313	335							
18	428	419							

Table 5
Comparison between the new method and the Sandwich method

	dimension of the problem					
	5	8	10	5	8	10
Z_1	133	290	439	T_1 301	769	1257
Z_2	141	231	337	T_2 161	461	733

Z_1 is the number of iterations for the new method with integer labelling.

Z_2 is the same with vector labelling.

T_1 is the total number of iterations for the Sandwich method (integer labelling).

T_2 is the number of genuine iterations for the Sandwich method.

The final gridsize of the new method is $n \times 3^{11}$.

The final gridsize of the Sandwich method is $4n \times 3^{10}$.

with $x_i = 0$ and $x_{i+1(\mathrm{mod}\, n)} > 0$. Moreover, an interior point x of S^n receives the label i if $g_i(x)/w_i \geq g_k(x)/w_k$ for all k, where w_j is the total supply of good j and $g_j(x)$ is its total demand, $j = 1, 2, \ldots, n$. This labelling rule is analoguous to rule 3 of MacKinnon [9]. Using vector labelling a point x on the boundary receives the label $e(i)$ if i is the lowest index with $x_i = 0$ and $x_{i+1(\mathrm{mod}\, n)} > 0$, whereas an interior point x receives the label $g(x)$. We define a t-dimensional face to be completely labelled if $\sum_{i=0}^{t} \lambda_i g(w^j) + \sum_{j=1}^{n-t-1} \mu_j e(\pi_j) = w$ has some non-negative solution λ_i^*, $i = 0, \ldots, t$, and μ_j^*, $j = 1, \ldots, n-t-1$. In [11] it is proved that a completely labelled subsimplex yields a good approximation of a fixed point.

In Table 5 we compare our results with those of Kuhn and MacKinnon [6].

7. Conclusions

In his paper [9] MacKinnon shows that for the Sandwich method the number or iterations depends highly on the factor of incrementation. He concludes that for integer labelling the factor of incrementation has to be rather low, e.g. between two and five. Using our method for integer labelling, we found that the best factor of incrementation is four, three and two for respectively the examples 1, 2 and 3, which agrees with the conclusions of MacKinnon. The situation is very different for vector labelling. Then the best factors we found were respectively five, five and eight. By making the choice of the new gridsize dependent on the difference of the last two approximations, computational experiences show that the factor of incrementation could be increased very fast. This is caused by the differentiability of the functions, which agrees with conclusions of MacKinnon (private communication). Observe that in the homotopy method the factor of incrementation must be always equal to two.

However, recently Saigal [10] developed a method, which allows a higher factor of incrementation if the function satisfies some strong conditions. Comparing our results with these of Wilmuth [16], our method takes for integer as well as for vector labelling significantly fewer iterations to reach a given level of gridrefinement than the Sandwich method and the homotopy method.

Moreover, the results for vectorlabelling can be improved by taking $x^* = \sum_{i=1}^{n} \lambda_i^* w^i$ as the new starting point. Furthermore, in case 3 problems (see [9]) the new algorithm can be expected to work better when extrapolative restarts are used.

Finally, it should be mentioned that it is very simple to implement the algorithm on the computer. In a subsequent paper [6] we reported about the generalization of the algorithm for point to set mappings in R^n.

References

[1] L.E.J. Brouwer, "Über eineindeutige, stetige Transformationen von Flächen in sich", *Mathematische Annalen* 67 (1910) 176–180.

[2] B.C. Eaves, "Computing Kakutani fixed points", *SIAM Journal on Applied Mathematics* 21 (1971) 236–244.

[3] B.C. Eaves, "Homotopies for computation of fixed points", *Mathematical Programming* 3 (1972) 1–22.

[4] B.C. Eaves and R. Saigal, "Homotopies for computation of fixed points on unbounded regions", *Mathematical Programming* 3 (1972) 225–237.

[5] H.W. Kuhn and J.G. MacKinnon, "Sandwich method for finding fixed points", *Journal of Optimization Theory and Applications* 17 (1975) 189–204.

[6] G. van der Laan and A.J.J. Talman, "A restart algorithm without an artificial level for computing fixed points on unbounded regions", Department of Actuarial Sciences and Econometrics, Tech. Rept. No. 31, Free University, Amsterdam, The Netherlands (1978).

[7] H.J. Lüthi, *Komplementaritäts- und Fixpunktalgorithmen in der mathematischen Programmierung, Spieltheorie und Ökonomie* (Springer, Berlin, 1976).

[8] O.H. Merrill, "Applications and extensions of an algorithm that computes fixed points of certain upper semi-continuous point to set mappings", Ph.D. Thesis, University of Michigan, Ann Arbor, MI. (1972).

[9] J.G. MacKinnon, "Solving economic general equilibrium models by the Sandwich method", in: S. Karamardian ed., *Fixed points algorithms and applications* (Academic Press, New York, 1977) pp. 367–398.

[10] R. Saigal, "On the convergence rate of algorithms for solving equations that are based on methods of complementary pivoting", *Mathematics of Operations Research* 2 (1977) 108–124.

[11] H.E. Scarf, "The approximation of fixed points of a continuous mapping", *SIAM Journal on Applied Mathematics* 15 (1967) 1328–1343.

[12] H.E. Scarf, *The computation of economic equilibria* (Yale University Press, New Haven, CT., 1973).

[13] L.S. Shapley, "On balanced games without side payments", in: T.C. Hu and S.M. Robinson eds., *Mathematical programming* (Academic Press, New York, 1973) pp. 261–290.

[14] M.J. Todd, *The computation of fixed points and applications* (Springer, Berlin, 1976).

[15] M.J. Todd, "Fixed point algorithms that allow restarting without an extra dimension", School of Operations Research and Industrial Engineering, Tech. Rept. No. 379, Cornell University, Ithaca, NY (1978).

[16] R.J. Wilmuth, "A computational comparison of fixed point algorithms which use complementary pivoting", in: S. Karmardian, ed., *Fixed points algorithms and applications* (Academic Press, New York, 1977) pp. 249–280.